ARI Research Note 87-48

DTIC FILE COPY

AD-A189 232

EXPERIMENTAL STUDIES OF NOVICE COMPUTER USERS

Donald J. Foss
University of Texas

for

Contracting Officer's Representative
Michael Drillings

BASIC RESEARCH LABORATORY
Michael Kaplan, Director

U. S. Army

Research Institute for the Behavioral and Social Sciences

October 1987

Approved for public release; distribution unlimited.

87 11

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

Research accomplished under contract
for the Department of the Army

University of Texas at Austin

Technical review by

Michael Kaplan

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ARI Research Note 87-48	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Experimental Studies of Novice Computer Users		5. TYPE OF REPORT & PERIOD COVERED Final Report April 82 - April 85 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald J. Foss		8. CONTRACT OR GRANT NUMBER(s) MDA903-82-C-0123
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Texas at Austin P.O. Box 7726 Austin, Texas 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q161102B74F
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		12. REPORT DATE October 1987 13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) - -		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE n/a

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

- -

18. SUPPLEMENTARY NOTES

Michael Drillings, contracting officer's representative

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Learning
Man-Machine,
Surrogate Models,
Transfer of Training

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research note examines the dynamics of the changing goal structure of a novice learning and carrying out computer-based tasks. Novices were provided with either a surrogate model for learning text editing, two versions of learning manual syntax, or two versions of manual organization. Results showed no effects due to surrogate models, but significant improvements in learning due either to more concrete learning model syntax, or a layered manual organization. Contradictory effects on training transfer were found.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

EXPERIMENTAL STUDIES OF NOVICE COMPUTER USERS

<u>SCIENTIFIC OBJECTIVES</u>: It is widely believed that the learning and effective use of complex systems, including many computer-based systems, requires an understanding of the principles behind the operation of those systems. Our main purpose was to investigate the savings that actually can be obtained by giving learners a "model" of the system in advance of training on it. A subsidiary goal was to examine the nature of "mental models" so that we could better understand how to construct useful ones. A third goal was to examine the principles that govern optimal sequencing of instructions when subjects learn command languages associated with such systems as text editors.

The initial focus of our work was on novice users, but that focus shifted during the course of the project. We adopted a perspective common to many workers in this area, namely we analyzed our learners' behavior in terms of goals and subgoals. One of our aims was to look at the dynamics of the changing goal structure as a novice both learned and carried out computer-based tasks. (These generally were either text editing or data-base manipulations.) Our initial studies on learning a text editor found significant differences in learning times among our novice subjects as we manipulated command sequencing and command syntax, but very little effect due to our "mental model." This led us to shift our attention to transfer of training more generally. Our objective was to find a metric for predicting where and when learners will benefit and where and when they will have difficulties as they move from one computer-based system to another. Another objective of this work was to develop a means for choosing between alternative designs for a new system when we know the background of its prospective users.

<u>SUMMARY</u>. In this section I will briefly report on the major projects carried out, as well as allude to some others. Two of the major projects are spelled out in

considerable detail in the accompanying papers.

We began by believing that we could substantially aid novice computer users to learn a text editor by giving them in advance an appropriately structured organizational framework (sometimes called a mental model or, following the terminology of Young, a surrogate model). Accordingly we carried out two major studies in which novices were taught a text editor called SOS. Both epxeriments are described in detail in the paper appearing in the *International Journal of Man-Machine Studies* (attached). Here I will very briefly describe the second and more extensive study. In that work we manipulated factorially three variables: (1) Whether or not subjects were given a metaphorically-based surrogate model to aid them in understanding the operating system and the text editor, and to aid them in problem solving and generalization tasks; (2) the syntax of the commands presented in the manual (some subjects got the "abstract" command syntax of the orginal manual, others got a more "concrete" version); and (3) the manual organization (some subjects got a "layered" version of the manual, others got the original version in which many ways of carrying out a function were introduced at once). Nine subjects were tested in each of the eight conditions of the experiment. Each subject participated for approximately three to four hours.

We collected time-stamped keystroke data, errors, verbalizations, and a number of other variables. After coding the data we had a very large number of dependent variables (e.g., times to use various commands, times to successfully use them, errors of many types) which we collapsed according to a rational scheme we devised, yielding an acceptable number of variables for multivariate analysis. In all the analyses we partialled out the effects of our subjects' SAT scores, a variable that was uniformly highly significant as one might expect.

We found significant effects for two of the three independent variables taken alone: manual syntax and manual organization. The effect due to the surrogate model or metaphor was not significant. Some effects were fairly substantial. For example, subjects in the better syntax conditions made one-fourth as many errors of one particular type as did subjects in the conditions with the poorer syntax. Also, 20%

fewer action commands were issued in the best condition relative to the poorest one.

Analyses were carried out on a variety of dependent variables that we thought would be affected by each of the independent variables. Suffice it to say here that we did not find a significant effect of the surrogate model on the dependent variables that we thought would be most affected by it. We did find, though, that the locus of the effects -- the dependent variables affected -- varied somewhat between the two significant independent variables. To a first approximation, the manual organization variable appeared to have its major influence on the subjects' planning operations. That is, the organization of the manual more strongly influenced the choice of command. On the other hand, the syntax variable appeared to have its influence primarily on the execution phase, after the selection of the action type had occurred. (This dichotomy was admittedly post hoc, devised after looking at the data.) As we said in our paper, "In general, then, the organization and syntax manipulations appear to affect different dependent variables. Their effects are localized. The original manual's organization and command format (syntax) both cause delays and a greater number of commands. Perhaps they also lead to similar feelings of frustration. But they do it by engendering more-or-less distinctive errors on the part of the subjects, causing their effects to be additive rather than multiplicative."

We also took a different approach -- via reading times -- to the question of how and when the differences in manual syntax and organization affect comprehension and performance. In one study we put the manuals on the computer and had subjects read them on line. The manuals were presented a line at a time (only one line was visible to the subject at any given time) in a self-paced fashion. That is, subjects pressed a key on the terminal each time they wanted to read a new line. Each line ended at a clause boundary. We collected the reading times for each of the lines. The two manuals we tested were identical on each line except for a few on which the commands were presented and demonstrated via examples. In one case the information on these "critical" lines was presented in the concrete syntax and with few different ways to carry out the command. In the other case the more "abstract" syntax and multiple ways of carrying out the command were presented. We looked

at average reading time per line (both for the lines in common and the critical lines). Subjects were also given a few editing tasks to carry out, which constituted a very clear test of comprehension. If subjects can carry out a task making direct use of the information just presented, then it seems quite clear that they have comprehended that information. We measured the errors in editing as well as the amount of time it took to use the commands correctly and the total time to complete the editing tasks

We found that subjects spent longer looking at the critical lines when they consisted of materials built from the abstract syntax than when they read lines with the concrete syntax (though this effect was consistent, it was not large and was not statistically significant). We also found a significant positive correlation between reading time per line and the time to successfully use the more complex commands. The correlation was .30 ($p < .05$). The analogous correlation was very small and very far from significant when we looked at reading time and time to use the simpler commands. Thus, the reading time data were consistent with the learning data from the earlier work, but the effects were disappointingly small. This may have been due to the fact that we were looking at novices. On some lines they may have skimmed over material when they were not at all sure what it meant, thereby reducing the reading time effects. This may work better if experienced (but not necessarily expert) users are the subjects.

We also completed a study with novices in which we gathered thinking aloud protocols as well as keystroke records. We found in our thinking aloud protocols that we could make good estimates of the subjects' moment by moment goals and subgoals (at least we were confident that we were generally correct in our assessments). We believe that this technique permitted us to observe the erroneous subgoals the learners set up. After examining a set of them, we concluded that it was highly unlikely that a traditional surrogate model would permit our learners to avoid these faulty subgoals, a conclusion consistent with the data reported above.

Before returning to the main issue of transfer, we conducted a project looking at the memory dynamics associated with the use of goals and subgoals in a data-base

system. Here we were concerned with aspects of the execution of a goal and how that related to demands on working memory.

To examine the interaction of memory limitations and problem solving goals we taught a set of novice subjects a data-base system called Omnidata. Earlier we had made an analysis of a set of tasks that can be carried out with Omni and we set up the goals and the subgoals for the subjects in the study. We assume that setting up the appropriate subgoals is a critical part of problem solving in a new domain, but for the purposes of our study we chose to finesse that aspect of the novice's task. Instead, we concentrated on questions concerning the way in which subgoal information is stored and retrieved while novices are carrying out tasks. A series of models was developed (these are described in detail in T. Kanarski's thesis) but only two will be touched on here. Consider a subject who is asked to delete information from a data base. Suppose that the subject -- who is given the overall plan about how to accomplish the task -- places the first subgoal into a working memory and then carries out the detailed instructions associated with that subgoal. Now consider what happens when the subgoal is carried out. On the one hand we might propose that the subgoal is deleted from working memory -- this would then be analogous to the situation in motor performance described by Sternberg, Monsell, Knoll, and Wright (1980). At the planning level, on the other hand, we might propose that the same subgoal is likely to be carried out again and again. In that case, a destructive read from working memory would be inefficient since, presumably, it would entail another search of long term memory, or the reconstruction of the same plan. Thus, let us suppose that novice subjects, as well as more experienced ones, keep in working memory subgoals that have just been activated. In that case, we will expect different behavior from the subjects than in the case where use of the subgoal removes it from the working memory. (We call these "nondestructive memory read" and "destructive memory read," respectively.)

In the experiment novice Omni users were given various tasks to accomplish and we measured the time subjects took to give each command. The clock was started at the end of the previous command. We found statistically reliable evidence in favor of the nondestructive read hypothesis. These results were reported at a meeting of the

Cognitive Science Society.

Returning to our original theme, that having to do with surrogate models, recall that we found in our experiments that the savings in learning time, errors, etc. due to the presence of such a model were not large nor even significant. This led to a shift in the focus of our work. A model should provide the basis for positive transfer: from the model to the target system. While the transfer effects we observed were decidedly minimal, we noted that there is plenty of anecdotal evidence (since demonstrated clearly in experimental settings) that the transfer between related systems such as two text editors should be positive. We then began to look at how to predict transfer from one complex system (e.g., a text editor or data-base system) to another.

Basically, our aims were (1) to find a metric for predicting the type and amount of transfer between systems, and (2) to develop a means for choosing between alternative designs for a new system when we know the background of its prospective users.

In order to examine these issues we designed a transfer of training study. This study was similar in spirit to those being conducted at about the same time by Singley and Anderson at Carnegie-Mellon, and by Kieras and Polson at Colorado, although we were not aware of their work until later. In our experiment the subjects all studied and answered questions about a screen-oriented text editor that is commonly used on DEC equipment, the K52 editor. One-third of the subjects had previously learned (part of) another commonly used screen editor, EMACS; one-third had learned (part of) a line-oriented editor, SOS; and one-third learned no editor but had equivalent amounts of hands on experience with the terminal -- they were taught the rudiments of the BASIC programming language.

As mentioned, subjects in the transfer experiment were given a common final task, that of learning some aspects of the K52 Editor available on our DEC computer. Training on it was provided by a series of 14 lessons, recorded on tape. After each lesson a "knowledge check" was given; this test examined subjects command of the

material just presented and also required some integration across lessons. Each knowledge check was five pages long. The items on the quizzes fell into one of three categories, in general. (1) Some items asked for descriptive information about particular commands or sets of commands. (2) Other items asked subjects to predict how a given file would appear after a particular command was executed. A copy of a short text file was provided, with cursor location marked, and subjects were asked to modify it appropriately. (3) And some items required subjects to apply commands to given situations. Some of these asked the subjects to tell what the next operation should be, while others required them to state a sequence of operations.

The data were coded so that various types of responses (and errors) could reliably be counted, and so that we could tell whether subjects understood both the main effects of commands and the "side effects" that they have on such things as cursor position.

The average time that subjects took to complete each of the 14 quizzes was measured. Informally, one might expect that subjects who had learned a screen editor on the first day would be faster than the other subjects on the K52 (new screen editor) quizzes. Results like that have been reported by the above-mentioned researchers. In our data, though, these subjects were significantly slower. The average time per quiz for the Basic, SOS, and EMACS groups were 301 sec, 303 sec, and 345 sec, respectively. The EMACS group did worse, according to an analysis carried out over the quizzes ($p<.01$). One reasonable interpretation of this finding is that our subjects did not know EMACS very well and they got confused on the tests because of the similarities to the K52 editor. To push it a step further, the reason for the confusion is that the planning stages of what to do with the screen editors are highly similar, while the syntax of the commands are somewhat different. Also, the effects of the commands on the cursor movements and the file is similar, but the exact way in which those effects are implemented differs in the two cases. Thus, the effects of learning one system on the transfer to another one may be dependent both on the similarities between the two systems at the "higher level" planning stages as well as at "lower level" stages involved with assembling the syntax of the command. Similarities at the higher levels may mediate the effects of similarities at lower levels.

With respect to the subjects' accuracy, the pattern of results shifts somewhat. The subjects who were exposed to both the SOS and the EMACS editors did better overall than did the subjects who learned the rudiments of BASIC. The overall percent correct for the BASIC, SOS, and EMACS groups was 57%, 65%, and 67%, respectively. The former differed significantly from the latter two. This pattern of results is consistent with the "high level," "low level" analysis, suggesting that the two initial editors had enough in common with K52 at the higher level so that the subjects who learned them could do well on the transfer task.

Certain questions arise naturally from these data. For example, it is unfortunately not clear to what extent the BASIC group was aided by their training since we did not have a No Training control. It will make a substantial difference in one's thinking if such a group would have scored 50% or 20% on the transfer task. Also, it there is reason to suspect that our results might have been different if our subjects had been more thoroughly trained. This last is an important point. Singley and Anderson found vaying degrees of postive transfer and no evidence for negative transfer (similar results have been reported by Polson and Kieras). Our somewhat diparate findings with the time measure may have been due to the fact that our subjects were not very well trained on the initial systems. Of course, there are also substantial methodological differences between the studies as well.

In the chapter discussing this work, we also raised a number of issues having to do with choosing between alternative designs for systems, and discussed the importance of transfer of training in developing a metric for making these choices. We argued that it is extraordinarily unlikely that we will be able to design in advance an optimal system because of the lack of a theory to guide such a design. We suggested that it might be easier (though still very difficult) to make a reasoned choice among alternative designs on a theoretically-grounded basis. In order to do this a metric for making such choices is required. We proposed that studies of transfer of training might help to identify the relevant dimensions along which an evaluation metric could be defined. Transfer might fruitfully be conceptualized by keeping distinct the notions of goal structure and command syntax. Such a

distinction may be useful, we said, because transfer may be differentially affected by these two. With the external task held constant, the goal structure may transfer more-or-less intact (yielding positive transfer) while changes in the syntax of commands and exactly what each accomplishes may lead to negative transfer. Thus, a transfer metric might do well to take into account both kinds of differences between the old and new systems. To date, this has not been accomplished, however.

CONCLUSIONS. We began by suggesting that surrogate models would yield large amounts of positive transfer and that it would be possible to state principles according to which one could build such models. The early experiments we conducted on learning test editors found significant effects due to the syntax and manual organization variables, but no effects due to the surrogate models. (The organization and syntax effects we reported have made their way into B. Schneiderman's recent text on design principles for human interfaces.) The organization of a training manual significantly affected performance, with the "layered" manual leading to better scores than the non-layered one. (Layering means that the modal way of carrrying out a function is first taught alone, later other ways are introducted; nonlayered means that the alternatives are presented in the same section of the manual). We also found that abstract syntax led to more errors in performance than did concrete syntax. Interestingly, these variables appeared to influence different dependent variables, one set having to do with planning and the other with execution of the plans. The effect of our surrogate model was negligible. These findings were corroborated by some work we did on evaluating learner's momentary goals using the "thinking aloud" technique. We found in our thinking aloud protocols that we could make good estimates of the subjects' moment by moment goals and subgoals, and observe the erroneous subgoals they set up. We concluded that it was unlikely that a traditional surrogate model would permit our learners to avoid the faulty subgoals.

In another line of work we found evidence that goals reside in working memory after being executed, e.g., the "nondestructive read" hypothesis garnered support. This work examined the relation between working memory and the dynamics of

learner's goal structures.

We noted that studying the effects of surrogate models is merely a special case of the general problem of transfer of training. Our work then moved explicitly into that area. We examined transfer between editors of different types and found some evidence for both positive and negative transfer. These results were interpreted in light of the distinction we've made between planning and execution of goals. Since others have not found much evidence for negative transfer (and ours was restricted to a time measure and not to a performance measure), we must be very cautious about that finding. (We noted that it may be a function of degree of training as well as of methodological differences between the studies.) However, the distinction between planning and execution of goals did account for data both in the original learning work as well as in the transfer work. It is clear that considerable work remains to be done in the transfer paradigm. Our chapter (enclosed) discusses further some theoretical issues that can potentially be powerfully addressed using the transfer paradigm.